

UJIAN AKHIR SEMESTER

“Pemrograman”

**Dosen Pengampu :
M. Fairul Filza, S.Kom,M.Kom**



Disusun oleh:

Nama : Muhammad Rifki

NIM : 22.11.4636

Kelas : IF 02

UNIVERSITAS AMIKOM YOGYAKARTA

2022/2023

1. Buatlah contoh program dari setiap bab materi pemrograman di bawah ini, beserta komentar pada setiap kode program yang Anda buat.

✓ Encapsulation

Code :

```
using System;

2 references
class TicketOrder
{
    private string name;
    private string nik;
    private string phoneNumber;
    private string vaccineCardNumber;
    private string address;

    // Getter dan Setter untuk masing-masing atribut
    1 reference
    public string Name
    {
        get { return name; }
        set { name = value; }
    }

    1 reference
    public string Nik
    {
        get { return nik; }
        set { nik = value; }
    }

    1 reference
    public string PhoneNumber
    {
        get { return phoneNumber; }
        set { phoneNumber = value; }
    }
}
```

```

1 reference
public string VaccineCardNumber
{
    get { return vaccineCardNumber; }
    set { vaccineCardNumber = value; }
}

1 reference
public string Address
{
    get { return address; }
    set { address = value; }
}

// Metode untuk melakukan pemesanan tiket
1 reference
public void PlaceOrder()
{
    // Implementasi logika untuk pemesanan tiket
    Console.WriteLine("Ticket order placed successfully!");
}
}

```

```

0 references
class Program
{
    0 references
    static void Main()
    {
        // Penggunaan kelas TicketOrder dengan encapsulation
        TicketOrder ticketOrder = new TicketOrder();
        ticketOrder.Name = "Gatotkaca";
        ticketOrder.Nik = "123456789";
        ticketOrder.PhoneNumber = "081234567890";
        ticketOrder.VaccineCardNumber = "VAX123";
        ticketOrder.Address = "123 Main St, City";

        ticketOrder.PlaceOrder();
    }
}

```

Running :

```
Ticket order placed successfully!
```

✓ Inheritance

Code :

```
using System;

// Kelas induk (superclass)
3 references
class Vehicle
{
    protected string brand;
    protected string color;

    // Konstruktor kelas Vehicle
    1 reference
    public Vehicle(string brand, string color)
    {
        this.brand = brand;
        this.color = color;
    }

    // Metode virtual untuk mencetak informasi kendaraan
    3 references
    public virtual void PrintInfo()
    {
        Console.WriteLine("Brand: " + brand);
        Console.WriteLine("Color: " + color);
    }
}
```

```

// Kelas turunan (subclass) yang mewarisi dari Vehicle
3 references
class Car : Vehicle
{
    private int numberOfDoors;

    // Konstruktor kelas Car
    1 reference
    public Car(string brand, string color, int numberOfDoors) : base(brand, color)
    {
        this.numberOfDoors = numberOfDoors;
    }

    // Implementasi ulang metode PrintInfo dari kelas Vehicle
    3 references
    public override void PrintInfo()
    {
        base.PrintInfo();
        Console.WriteLine("Number of doors: " + numberOfDoors);
    }
}

```

```

class Program
{
    0 references
    static void Main(string[] args)
    {
        // Membuat objek Car
        Car myCar = new Car("Toyota", "Red", 4);

        // Memanggil metode PrintInfo pada objek Car
        myCar.PrintInfo();

        Console.ReadLine();
    }
}

```

Running :

```

Brand: Toyota
Color: Red
Number of doors: 4

```

✓ Polymorphism

Code :

```

using System;

// Membuat kelas abstrak Vehicle
4 references
abstract class Vehicle
{
    // Mendefinisikan metode abstrak untuk menghitung kecepatan
    6 references
    public abstract void CalculateSpeed();
}

// Membuat kelas Car yang mewarisi dari kelas Vehicle
3 references
class Car : Vehicle
{
    // Implementasi metode CalculateSpeed dari kelas Vehicle
    4 references
    public override void CalculateSpeed()
    {
        Console.WriteLine("Kecepatan mobil dihitung.");
    }
}

// Membuat kelas Bicycle yang mewarisi dari kelas Vehicle
3 references
class Bicycle : Vehicle
{
    // Implementasi metode CalculateSpeed dari kelas Vehicle
    4 references
    public override void CalculateSpeed()
    {
        Console.WriteLine("Kecepatan sepeda dihitung.");
    }
}

```

```

// Membuat kelas Program sebagai kelas entry point
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        // Membuat objek Car dan Bicycle
        Car car = new Car();
        Bicycle bicycle = new Bicycle();

        // Memanggil metode CalculateSpeed pada objek Car
        car.CalculateSpeed();

        // Memanggil metode CalculateSpeed pada objek Bicycle
        bicycle.CalculateSpeed();

        // Menggunakan polymorphism dengan tipe Vehicle
        Vehicle vehicle1 = new Car();
        Vehicle vehicle2 = new Bicycle();

        // Memanggil metode CalculateSpeed pada objek yang memiliki tipe Vehicle
        vehicle1.CalculateSpeed();
        vehicle2.CalculateSpeed();

        // Menunggu input pengguna sebelum keluar dari program
        Console.ReadLine();
    }
}

```

Running :

Kecepatan mobil dihitung.
Kecepatan sepeda dihitung.
Kecepatan mobil dihitung.
Kecepatan sepeda dihitung.

✓ Abstraction

Code :

```
using System;

// Abstraksi adalah konsep dalam pemrograman berorientasi objek (OOP) yang
// memungkinkan kita untuk menyembunyikan detail implementasi dan hanya
// mengekspos fungsionalitas yang relevan.
// Dalam contoh ini, kita akan membuat sebuah abstraksi dari bentuk geometri
// yang disebut "Bangun" (Shape) dan implementasinya dengan beberapa bentuk
// konkrit seperti "Persegi" (Square) dan "Lingkaran" (Circle).

// Langkah 1: Membuat kelas abstrak "Bangun" (Shape)
2 references
public abstract class Shape
{
    // Metode abstrak "HitungLuas" harus diimplementasikan oleh semua kelas turunannya
    4 references
    public abstract double HitungLuas();
}

// Langkah 2: Membuat kelas turunan dari "Bangun" (Shape), yaitu "Persegi" (Square)
3 references
public class Square : Shape
{
    private double sisi;

    // Konstruktor untuk menginisialisasi sisi persegi
    1 reference
    public Square(double sisi)
    {
        this.sisi = sisi;
    }
}
```

```
// Langkah 4: Membuat kelas utama "Program" untuk menjalankan program
0 references
public class Program
{
    0 references
    public static void Main(string[] args)
    {
        // Buat objek persegi
        Square persegi = new Square(5);
        // Hitung luas persegi menggunakan metode "HitungLuas" yang diwarisi dari "Bangun" (Shape)
        double luasPersegi = persegi.HitungLuas();
        Console.WriteLine("Luas Persegi: " + luasPersegi);

        // Buat objek lingkaran
        Circle lingkaran = new Circle(3);
        // Hitung luas lingkaran menggunakan metode "HitungLuas" yang diwarisi dari "Bangun" (Shape)
        double luasLingkaran = lingkaran.HitungLuas();
        Console.WriteLine("Luas Lingkaran: " + luasLingkaran);

        // Tunggu hingga pengguna menekan tombol sebelum keluar
        Console.ReadLine();
    }
}
```

```

using System;

0 references
class Program
{
    0 references
    static void Main()
    {
        Console.WriteLine("=== Pemesanan Tiket Pesawat ===");
        Console.WriteLine();

        // Input informasi pemesanan
        Console.Write("Masukkan Nama: ");
        string nama = Console.ReadLine();

        Console.Write("Masukkan NIK: ");
        string nik = Console.ReadLine();

        Console.Write("Masukkan Nomor Telepon: ");
        string nomorTelepon = Console.ReadLine();

        Console.Write("Masukkan Nomor Kartu Vaksin: ");
        string nomorKartuVaksin = Console.ReadLine();

        Console.Write("Masukkan Alamat Rumah: ");
        string alamatRumah = Console.ReadLine();
    }
}

```

Running :

```

        return sisi * sisi;
    }
}

// Langkah 3: Membuat kelas turunan lainnya dari "Bangun" (Shape), yaitu "Lingkaran" (Circle)
3 references
public class Circle : Shape
{
    private double radius;

    // Konstruktor untuk menginisialisasi jari-jari lingkaran
    1 reference
    public Circle(double radius)
    {
        this.radius = radius;
    }

    // Implementasikan metode "HitungLuas" sesuai dengan rumus luas lingkaran
    2 references
    public override double HitungLuas()
    {
        return Math.PI * radius * radius;
    }
}

```

Luas Persegi: 25

Luas Lingkaran: 28.2743338823081

2. Misalkan Anda bekerja di perusahaan transportasi. Perusahaan tersebut meminta Anda untuk membuat aplikasi untuk proses pemesanan tiket, dengan kualifikasi: setiap pengguna yang memesan tiket transportasi harus menyebutkan nama, nomor induk (NIK), nomor telepon, nomor kartu vaksin, dan alamat rumah.

Buatlah programnya sesuai dengan penjelasan di atas. Tunjukkan hasil program dalam bentuk screenshot dan jelaskan cara menggunakan program tersebut.

Code :


```

        Console.WriteLine();
        Console.WriteLine("=== Informasi Pemesanan ===");
        Console.WriteLine("Nama: " + nama);
        Console.WriteLine("NIK: " + nik);
        Console.WriteLine("Nomor Telepon: " + nomorTelepon);
        Console.WriteLine("Nomor Kartu Vaksin: " + nomorKartuVaksin);
        Console.WriteLine("Alamat Rumah: " + alamatRumah);

        Console.WriteLine();
        Console.WriteLine("Terima kasih telah melakukan pemesanan tiket pesawat!");
    }
}

```

Running :

```

=== Pemesanan Tiket Pesawat ===
Masukkan Nama: Aku
Masukkan NIK: 320239302020
Masukkan Nomor Telepon: 0876543219
Masukkan Nomor Kartu Vaksin: 234
Masukkan Alamat Rumah: Jl.Hidaytullah No.20
=== Informasi Pemesanan ===
Nama: Aku
NIK: 320239302020
Nomor Telepon: 0876543219
Nomor Kartu Vaksin: 234
Alamat Rumah: Jl.Hidaytullah No.20

Terima kasih telah melakukan pemesanan tiket pesawat!

```

Penjelasan :

1. Program akan meminta Anda untuk memasukkan informasi pemesanan seperti nama, NIK, nomor telepon, nomor kartu vaksin, dan alamat rumah.
2. Ketikkan informasi yang diminta dan tekan tombol Enter setelah setiap input.
3. Setelah Anda memasukkan semua informasi, program akan menampilkan kembali informasi pemesanan yang telah Anda berikan.
4. Selesai! Program akan mengucapkan terima kasih dan berakhir.